

# Fire Effect

Presented by: Bryden, Alex and Terrence

# Fire effect tutorial coding

# Starting code

```
1 let system;
2
3< function setup() {
4   createCanvas(720, 400);
5   system = new ParticleSystem(createVector(width *2 , height));
6 }
7
8< function draw() {
9   background(51);
10  system.addParticle();
11  system.run();
12 }
13< let Particle = function(position) {
14   this.acceleration = createVector(0, -0.05);
15   this.velocity = createVector(random(-1, 1), random(-1, 0));
16   this.position = position.copy();
17   this.lifespan = 255;
18 };
19
20< Particle.prototype.run = function() {
21   this.update();
22   this.display();
23 };
24
25< Particle.prototype.update = function(){
26
27 };
28
29< Particle.prototype.display = function() {
30
31 };
32
33< Particle.prototype.isDead = function(){
34
35 };
36
37< let ParticleSystem = function(position) {
38
39 };
40
41< ParticleSystem.prototype.addParticle = function() {
42
43 };
44
45< ParticleSystem.prototype.run = function() {
46
47 };|
```

```
1 let system;
2
3 function setup() {
4   createCanvas(720, 400);
5   system = new ParticleSystem(createVector(width *2 , height));
6 }
7
8 function draw() {
9   background(51);
10  system.addParticle();
11  system.run();
12 }
13
14 let Particle = function(position) {
15   this.acceleration = createVector(0, -0.05);
16   this.velocity = createVector(random(-1, 1), random(-1, 0));
17   this.position = position.copy();
18   this.lifespan = 255;
19 }
20
21 Particle.prototype.run = function() {
22   this.update();
23   this.display();
24 }
```

```
24
25 Particle.prototype.update = function(){
26
27 };
28
29 Particle.prototype.display = function() {
30
31 };
32
33 Particle.prototype.isDead = function(){
34
35 };
36
37 let ParticleSystem = function(position) {
38
39 };
40
41 ParticleSystem.prototype.addParticle = function() {
42
43 };
44
45 ParticleSystem.prototype.run = function() {
46
47 };
```

```
Particle.prototype.update = function(){
    this.velocity.add(this.acceleration);
    this.position.add(this.velocity);
    this.lifespan -= 8;
};

Particle.prototype.display = function() {
    noStroke();
    colorMode(HSB);
    var hue = (120-this.lifespan) * 0.5;
    fill(hue, 255, 255, this.lifespan);
    ellipse(this.position.x, this.position.y, 12, 12);
};
```

```
Particle.prototype.isDead = function(){
    return this.lifespan < 0;
};

let ParticleSystem = function(position) {
    this.origin = position.copy();
    this.particles = [];
};
```

```
ParticleSystem.prototype.addParticle = function() {
    this.particles.push(new Particle(this.origin));
};

ParticleSystem.prototype.run = function() {
    for (let i = this.particles.length-1; i >= 0; i--) {
        let p = this.particles[i];
        p.run();
        if (p.isDead()) {
            this.particles.splice(i, 1);
        }
    }
};
```

Final Code:

```
1 let system;
2
3✓ function setup() {
4   createCanvas(720, 400);
5   system = new ParticleSystem(createVector(width *2 , height));
6 }
7
8✓ function draw() {
9   background(51);
10  system.addParticle();
11  system.run();
12 }
13✓ let Particle = function(position) {
14   this.acceleration = createVector(0, -0.05);
15   this.velocity = createVector(random(-1, 1), random(-1, 0));
16   this.position = position.copy();
17   this.lifespan = 255;
18 };
19
20✓ Particle.prototype.run = function() {
21   this.update();
22   this.display();
23 };
24
```

```
Particle.prototype.update = function(){
  this.velocity.add(this.acceleration);
  this.position.add(this.velocity);
  this.lifespan -= 8;
};

Particle.prototype.display = function() {
  noStroke();
  colorMode(HSB);
  var hue = (120-this.lifespan) * 0.5;
  fill(hue, 255, 255, this.lifespan);
  ellipse(this.position.x, this.position.y, 12, 12);
};

Particle.prototype.isDead = function(){
  return this.lifespan < 0;
};

let ParticleSystem = function(position) {
  this.origin = position.copy();
  this.particles = [];
};

ParticleSystem.prototype.addParticle = function() {
  this.particles.push(new Particle(this.origin));
};

ParticleSystem.prototype.run = function() {
  for (let i = this.particles.length-1; i >= 0; i--) {
    let p = this.particles[i];
    p.run();
    if (p.isDead()) {
      this.particles.splice(i, 1);
    }
  }
};
```

# Alex's Creative Portion

# Alex-Creative Portion

```
let system=[];
```

Makes the “system” variable an array at the start of the code.

```
function mouseDragged() {
  if (mouseY >= 100 && mouseY<=200) {
    system.push(new ParticleSystem(createVector(mouseX,
    mouseY)));
  }
}

for (var a = 0; a < system.length; a++){
  system[a].addParticle();
  system[a].run();
}
```

The first image creates a new instance of the particle system each time the mouse is clicked/dragged. This is passed into the loop where it adds a counter to the array to display additional fires.

# Alex-Creative Portion

```
inc = 240;  
bigfire=0;
```

Creates a global variables to apply to the entire code.

```
function mousePressed() {  
    inc+=1;  
    bigfire+=0.001;  
}
```

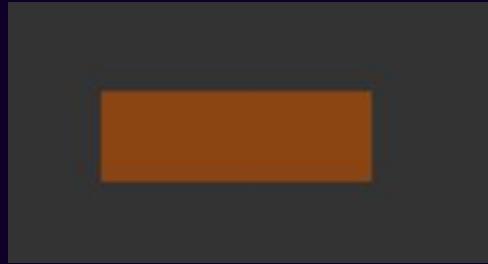
When the mouse is clicked it adds certain values to the variables created before.

```
this.acceleration = createVector(0, -0.05);  
this.velocity = createVector(random(-0.5-bigfire,  
0.5+bigfire), random(-0.1-bigfire, 0+bigfire));  
this.position = position.copy();  
this.lifespan = inc;  
};
```

Finally, the values are added to others to change things like the size and speed of the fire based on the number of clicks.

# Bryden's Creative Portion

# Starting off the code.



```
colorMode(RGB);
fill(139, 69, 19);
rect(125, 125, 150, 50);
```

```
let system;
let Thing;
let Thing2;
let Thing3;
let Thing4;
let Thing5;
let Thing6;
```

To start the code off I first displayed a Rectangle on the screen and added a Bunch of arrays to hold each fire particle with the different vectors and accelerations.

\*Commands and what they do will be displayed after explanation\*

# Adding functionality to the code.

What does this function do?

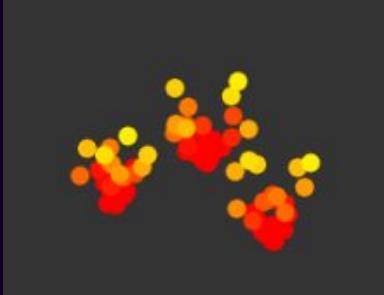
```
if (mouseY <= 175 && mouseIsPressed && mouseX <= 275 && mouseX >= 125 && mouseY >= 125){  
    if (Fireon == 0){  
        system = new ParticleSystem(createVector(140 , 135));  
        Thing = new ParticleSystem(createVector(180,135));  
        Thing2 = new ParticleSystem(createVector(230,135));  
        Thing3 = new ParticleSystem(createVector(260,135));  
        Fireon = 1;  
    }  
    else if (Fireon == 1){  
        system = new ParticleSystem(createVector(1000 , 135));  
        Thing = new ParticleSystem(createVector(1000,135));  
        Thing2 = new ParticleSystem(createVector(1000,135));  
        Thing3 = new ParticleSystem(createVector(1000,135));  
        Fireon = 0;  
    }  
}
```

What happens if you click it?



# More fire?

What does this do?



```
else
{
    if (x == 0){
        Thing4 = new ParticleSystem(createVector(mouseX, mouseY));
        x = 1;
    }
    else if (x == 1){
        Thing5 = new ParticleSystem(createVector(mouseX, mouseY));
        x = 2;
    }
    else if (x == 2){
        Thing6 = new ParticleSystem(createVector(mouseX, mouseY));
        x = 0;
    }
}
```

# Command Explanations

colorMode()

fill()

rect()

Arrays?

If statements.

# Terrence's Creative Portion



How does my code work?

# Terrence-Creative Portion

```
sketch.js*
1 // Declaring the global variable
2 let bubbles = [];
3
4 // Black Background
5 function setup() {
6   createCanvas(600,400);
7 }
8
9 // Draws the fire
10 function draw() {
11   background(0);
12
13   // Creates the bubbles
14   for (let i = 0; i < 7; i++) {
15     let b = new Bubble();
16     bubbles.push(b);
17   }
18
19 // Bubble Length
20 for (let i = bubbles.length-1; i >= 0; i--)
21   bubbles[i].update();
22   bubbles[i].show();
23
24   // Removes the bubbles
25   if (bubbles[i].finished()) {
26     bubbles.splice(i, 1);
27   }
28 }
29 }
```

```
sketch.js*
32 // Setting The fire's location/position
33 constructor() {
34   this.x = 300;
35   this.y = 400;
36   this.r = 30;
37   this.vx = random(-1, 1);
38   this.vy = random(-5, -1);
39   this.alpha = 255;
40 }
41
42 // Building the fire's base
43 update() {
44   this.x += this.vx;
45   this.y += this.vy;
46   this.alpha -= 7;
47   this.r -= 1;
48 }
49
50 // Displaying the fire
51 show() {
52   noStroke();
53   fill(13, 255, 0, 127);
54   ellipse(this.x, this.y, this.r)
55 }
56
57 // Keeps the Fire running
58 finished() {
59   return this.alpha < 0;
60 }
```

# A view of my Fire Effect

